

日本国特許庁  
JAPAN PATENT OFFICE

10/083572  
10/083572  
02/27/02  
J1017 U.S. PTO

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出願年月日  
Date of Application:

2001年 6月15日

出願番号  
Application Number:

特願2001-181791

出願人  
Applicant(s):

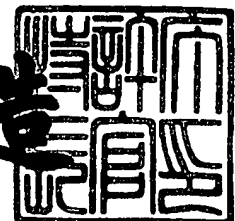
株式会社東芝

CERTIFIED COPY OF  
PRIORITY DOCUMENT

2001年12月28日

特許庁長官  
Commissioner,  
Japan Patent Office

及川耕造



出証番号 出証特2001-3112835

【書類名】 特許願

【整理番号】 A000102769

【提出日】 平成13年 6月15日

【あて先】 特許庁長官 殿

【国際特許分類】 G06F 9/00

【発明の名称】 分散システムおよび同システムの多重化制御方法

【請求項の数】 8

【発明者】

    【住所又は居所】 東京都府中市東芝町 1 番地 株式会社東芝府中事業所内

    【氏名】 遠藤 浩太郎

【特許出願人】

    【識別番号】 000003078

    【氏名又は名称】 株式会社 東芝

【代理人】

    【識別番号】 100058479

    【弁理士】

    【氏名又は名称】 鈴江 武彦

    【電話番号】 03-3502-3181

【選任した代理人】

    【識別番号】 100084618

    【弁理士】

    【氏名又は名称】 村松 貞男

【選任した代理人】

    【識別番号】 100068814

    【弁理士】

    【氏名又は名称】 坪井 淳

【選任した代理人】

    【識別番号】 100092196

    【弁理士】

【氏名又は名称】 橋本 良郎

【選任した代理人】

【識別番号】 100091351

【弁理士】

【氏名又は名称】 河野 哲

【選任した代理人】

【識別番号】 100088683

【弁理士】

【氏名又は名称】 中村 誠

【選任した代理人】

【識別番号】 100070437

【弁理士】

【氏名又は名称】 河井 将次

【手数料の表示】

【予納台帳番号】 011567

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【ブルーフの要否】 要

【書類名】 明細書

【発明の名称】 分散システムおよび同システムの多重化制御方法

【特許請求の範囲】

【請求項 1】 ネットワークで接続された  $n$  台のコンピュータを同期的に動作させる分散システムであって、少なくとも  $(n - f)$  台以上での多重化を保証する分散システムにおいて、

前記コンピュータそれぞれは、

各々が次に処理する候補として選択した入力データを前記ネットワークを介して収集する入力候補収集手段と、

前記入力候補収集手段により収集された入力データが  $(n - f)$  個以上存在する場合に、その中に同一内容の入力データが  $(n - f)$  個以上あるか否かを判定し、 $(n - f)$  個以上あったときに、その入力データを次に処理する対象として確定する第 1 の入力候補選定制御手段と、

前記収集された入力データの中に同一内容の入力データが  $(n - f)$  個以上なかったときに、前記収集された入力データ数の過半数を占める同一内容の入力データが存在するか否かを判定し、存在したときに、その入力データを自候補とするとともにそれ以外の他の候補の入力データをすべて破棄した上で前記入力候補収集手段に入力データの収集を再実行させる第 2 の入力候補選定制御手段と、

前記収集された入力データ数の過半数を占める同一内容の入力データが存在しなかったときに、前記収集された入力データの中からいずれかの入力データを任意に選択して自候補とするとともに、それ以外の他の候補の入力データをすべて破棄した上で前記入力候補収集手段に入力データの収集を再実行させる第 3 の入力候補選定制御手段と

を具備することを特徴とする分散システム。

【請求項 2】  $f$  は、 $3f < n$  を満たす最大の整数であることを特徴とする請求項 1 記載の分散システム。

【請求項 3】 前記コンピュータそれぞれは、

前記第 1 の入力候補確定制御手段により確定された入力データを保持するジャーナル手段と、

自コンピュータで既に確定済みである工程の入力データを他のコンピュータが収集しているときに、前記ジャーナル手段に保持された入力データを確定済みの入力データとして送信する第1の入力候補調整制御手段と、

前記入力候補収集手段による入力データの収集時に、他のコンピュータから確定済みの入力データが送信されたときに、その入力データを次に処理する対象として確定する第2の入力候補調整制御手段と

をさらに具備することを特徴とする請求項1または2記載の分散システム。

【請求項4】 前記ジャーナル手段は、前記入力データを最新のものから予め定められた工程数だけ保持し、

前記第1の入力候補調整制御手段は、他のコンピュータに送信すべき確定済みの入力データが前記ジャーナル手段に保持されていないときに、その旨を通知する手段を具備し、

前記コンピュータそれぞれは、

自コンピュータで既に確定済みである各工程での直前の状態を予め定められた工程分まで保持する状態保持手段と、

前記状態保持手段に保持された各工程での直前の状態を他のコンピュータとの間で送受信する状態送受信手段と、

前記入力候補収集手段による入力データの収集時に、その収集された入力データ数と他のコンピュータから確定済みの入力データが前記ジャーナル手段にも既に保持されていない旨を通知された数との和が $(n - f)$ 以上であって、前記収集された入力データ数が $(n - f)$ 未満であったときに、他のすべてのコンピュータの中で確定済みの工程が最も進んだ他のコンピュータにおける最新の確定済みの工程での直前の状態を前記状態送受信手段により取得して自コンピュータに複製するスキップ手段と

をさらに具備することを特徴とする請求項3記載の分散システム。

【請求項5】 ネットワークで接続された $n$ 台のコンピュータを同期的に動作させる分散システムであって、少なくとも $(n - f)$ 台以上での多重化を保証する分散システムの多重化制御方法であって、

各々が次に処理する候補として選択した入力データを前記ネットワークを介し

て収集する入力候補収集ステップと、

前記入力候補収集ステップにより収集された入力データが  $(n - f)$  個以上存在する場合に、その中に同一内容の入力データが  $(n - f)$  個以上あるか否かを判定し、 $(n - f)$  個以上あったときに、その入力データを次に処理する対象として確定する第 1 の入力候補選定制御ステップと、

前記収集された入力データの中に同一内容の入力データが  $(n - f)$  個以上なかったときに、前記収集された入力データ数の過半数を占める同一内容の入力データが存在するか否かを判定し、存在したときに、その入力データを自候補とするとともにそれ以外の他の候補の入力データをすべて破棄した上で前記入力候補収集ステップに入力データの収集を再実行させる第 2 の入力候補選定制御ステップと、

前記収集された入力データ数の過半数を占める同一内容の入力データが存在しなかったときに、前記収集された入力データの中からいずれかの入力データを任意に選択して自候補とするとともに、それ以外の他の候補の入力データをすべて破棄した上で前記入力候補収集手段に入力データの収集を再実行させる第 3 の入力候補選定制御ステップと

を具備することを特徴とする分散システムの多重化制御方法。

【請求項 6】  $f$  は、 $3f < n$  を満たす最大の整数であることを特徴とする請求項 5 記載の分散システムの多重化制御方法。

【請求項 7】 前記第 1 の入力候補確定制御ステップにより確定された入力データを保持するジャーナルステップと、

自コンピュータで既に確定済みである工程の入力データを他のコンピュータが収集しているときに、前記ジャーナルステップにより保持された入力データを確定済みの入力データとして送信する第 1 の入力候補調整制御ステップと、

前記入力候補収集ステップによる入力データの収集時に、他のコンピュータから確定済みの入力データが送信されたときに、その入力データを次に処理する対象として確定する第 2 の入力候補調整制御ステップと

をさらに具備することを特徴とする請求項 6 または 7 記載の分散システムの多重化制御方法。

【請求項 8】 前記ジャーナルステップは、前記入力データを最新のものから予め定められた工程数だけ保持し、

前記第 1 の入力候補調整制御ステップは、他のコンピュータに送信すべき確定済みの入力データが前記ジャーナル手段に保持されていないときに、その旨を通知するステップを具備し、

自コンピュータで既に確定済みである各工程での直前の状態を予め定められた工程分まで保持する状態保持ステップと、

前記状態保持ステップにより保持された各工程での直前の状態を他のコンピュータとの間で送受信する状態送受信ステップと、

前記入力候補収集ステップによる入力データの収集時に、その収集された入力データ数と他のコンピュータから確定済みの入力データが前記ジャーナルステップによっても既に保持されていない旨を通知された数との和が  $(n - f)$  以上であって、前記収集された入力データ数が  $(n - f)$  未満であったときに、他のすべてのコンピュータの中で確定済みの工程が最も進んだ他のコンピュータにおける最新の確定済みの工程での直前の状態を前記状態送受信ステップにより取得して自コンピュータに複製するスキップステップと

をさらに具備することを特徴とする請求項 7 記載の分散システムの多重化制御方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

この発明は、4 台以上のコンピュータがネットワークで接続された分散システムおよび同システムの多重化制御方法に係り、特に、スプリットブレインの防止と故障発生時におけるリアルタイム性の確保とを両立させることを可能とした分散システムおよび同システムの多重化制御方法に関する。

【0002】

【従来の技術】

近年、コンピュータ技術やネットワーク技術の向上は目覚ましく、これに伴って、業務の電算化が広く行われている。また、その業務の内容によっては、故障

などによる中断が許されないものも多く、最近では、複数のコンピュータをネットワークで結合した分散システムを構築することが一般的になりつつある。そして、この分散システムの運用手法の1つに、整列マルチキャストを用いた決定性のプログラムの実行の多重化が存在する。

【0003】

まず、「整列マルチキャスト」、「決定性のプログラム」および「多重化」について説明する。

【0004】

・ 整列マルチキャスト

複数のコンピュータが結合した分散システムのような環境では、各コンピュータが独立して動作する。したがって、これらのコンピュータを同期的に動作させるためには、特別な仕組みが必要である。整列マルチキャストは、分散システムへの入力をすべてのコンピュータに配送する仕組みであり、データの到着順序がすべてのコンピュータで同じであることを保証するものである。

【0005】

・ 決定性のプログラム

プログラムの実行は、コンピュータに入力が与えられると、その時のコンピュータの状態によって、出力と次の状態とを決めるものであると考えることができる。そして、決定性 (deterministic) のプログラムは、与えられた入力にしたがって、出力と次の状態とが一意的に決まるプログラムとして定義される。具体的には、不定値や乱数の参照等がないプログラムのことをいう。決定性のプログラムの特徴は、初期状態と入力列とが決まれば、その実行が一意的であることである。以下、本明細書でプログラムと称するとき、決定性のプログラムのことをさすものとする。

【0006】

・ 多重化

分散システムでは、各コンピュータが独立に故障する可能性がある。仮に、1つのコンピュータが故障しただけでシステム全体が機能しない場合は、分散システムの稼働率は、1台のコンピュータの稼働率よりも低くなってしまふ。かかる



事態を防止するために、システム全体に係わる処理は多重化することが必要である。逆に、多重化することによって、分散システムの稼働率を1台のコンピュータの稼働率よりも高くすることが可能である。たとえば、稼働率99パーセントのコンピュータ10台で構成する分散システムが、まったく多重化されていないとすると、その分散システムの稼働率は90%程度である。もし、これが多重化によって3台の故障まで耐え得るとすると、稼働率は、99.9998%程度となる。

## 【0007】

次に、整列マルチキャストを用いた決定性のプログラムの実行の多重化について説明する。ここでは、複数台のコンピュータによって構成される分散システムであって、多重化を構成するそれぞれのコンピュータが、同一のプログラムを有していると想定する。

## 【0008】

まず、すべてのコンピュータは、同一の初期状態からはじまる。その後、入力されるデータは、必ず整列マルチキャストを通して、すべてのコンピュータに同一順序で配送され、それぞれのプログラムが実行される。

## 【0009】

各プログラムへの入力列は、この整列マルチキャストにより、同一順序となっているので、決定性のプログラムの特徴により、すべてのコンピュータの状態が同一に保たれ、出力列もすべて同じとなる。つまり、プログラムの実行が多重化される。

## 【0010】

ここで、整列マルチキャストの実現方法について、その概要を説明する。

## 【0011】

特別なハードウェアによらずに整列マルチキャストを実現するためには、コンピュータ間で適切なアルゴリズムにしたがってメッセージをやり取りすること、つまりプロトコルが用いられる。アルゴリズムを具体的に説明する前に、注意すべき点を列挙する。

## 【0012】

すべてのコンピュータが、いつでも故障停止する可能性があることを前提としており、多重化として成立するためには、特定のコンピュータに全体の処理が依存してはならない。したがって、次のことに注意する必要がある。

【0013】

(1) 分散システムへの入力を受け付けを特定のコンピュータに固定しない。

【0014】

たとえば、特定のコンピュータに入力の受付を固定し、すべての入力をそのコンピュータにいったん転送することによって入力の順序を決定し、その順序で配送するといった単純なアルゴリズムは使えない。このアルゴリズムでは、入力受付を固定したコンピュータが故障停止すると、その時点で入力の順序が決定できなくなってしまう。

【0015】

(2) 入力の配送の完了まちあわせを特定のコンピュータに固定しない。

【0016】

たとえば、特定のコンピュータが、停止していないすべてのコンピュータに配送を行うようにする、といった単純なアルゴリズムは使えない。このアルゴリズムでは、配送コンピュータが配送の途中で故障停止してしまうと、一部のコンピュータにのみ配送されたまま、配送が完了しなくなってしまう。

【0017】

以上を踏まえて、前述のアルゴリズムを具体的に説明する。

【0018】

従来では、故障検出が重要な役割を果たす。典型的には、故障検出はハートビート・タイムアウト・アルゴリズムによって行われる。このアルゴリズムは、各コンピュータが定期的に出送するハートビート（心拍）が一定時間以上確認できない場合に、当該コンピュータの故障を判定するというものである。

【0019】

また、各コンピュータは、入力受付キューをもつ。第1ステップとして、それぞれのコンピュータは、入力受付キューの先頭にある入力をそのコンピュータにおける「入力候補」として他のすべてのコンピュータに配送する。また、入力受

付キューが空のコンピュータでは、他のコンピュータの第1ステップとして最初に得られた「入力候補」を自分の「入力候補」として他のすべてのコンピュータに配送する。

【0020】

第1ステップの最終的な結果として、各コンピュータは、すべてのコンピュータについて、「入力候補」を得るか、「故障検出」を得るか、または、その双方を得る。ここでは、すべてのコンピュータについての「入力候補」および「故障検出」の一覧を単に「一覧」と呼ぶことにする。

【0021】

第2ステップとして、それぞれのコンピュータは、自分の「一覧」を他のすべてのコンピュータに配送する。ここで注意すべき点は、これらの「一覧」が、各コンピュータごとに異なっているかも知れないということである。なぜなら、第1ステップの途中で故障停止が発生した場合には、「入力候補」が部分的にしか配送されていないかも知れない。また、第2ステップの開始の時点で、「故障検出」にはずれがあるかも知れないからである。

【0022】

第2ステップの結果として、各コンピュータは、他のコンピュータから得られた「一覧」が自分の「一覧」と異なっている場合、これらを合併して自分の「一覧」にし、第2ステップを繰り返し実行する。すると、この第2ステップの最終的な結果として、故障していない他のコンピュータがもつ「一覧」がすべて自分の「一覧」と一致する。この時点で、プロトコルは完了する。

【0023】

なお、整列マルチキャストとして配送される入力は、その「一覧」にある「入力候補」の中から各自が同一の決まったルールで選べばよい（たとえば先頭にあるもの）。そして、最後に、その選んだ入力を入力受付キューから取り除く。

【0024】

以上の手順により、複数のコンピュータをネットワークで結合した分散システムにおける、整列マルチキャストを用いた決定性のプログラムの実行の多重化が実現される。

【0025】

【発明が解決しようとする課題】

ところで、前述した手順では、次のような問題点があった。

【0026】

(1) スプリットブレイン

スプリットブレインは、実行のコンテキスト（状態）が2つ以上に分かれてしまうことをさす。このスプリットブレインは、故障検出が誤って行われたときに発生する。たとえば、システムを構成するコンピュータが、2つのコンピュータ群の間に互いに通信できない状態となった場合（ネットワークパーティショニング）、それぞれのコンピュータ群は、互いに故障検出し、独立して動作をはじめ。あるいは、一時的な高負荷のために、ハートビートの送受信が中断して故障の誤検出が発生し、スプリットブレインに陥る場合もある。

【0027】

多重化された処理は、システムの中で重要な処理であるはずである。ここでスプリットブレインが起きると、その処理に一貫性がなくなり、システム全体に致命的な影響を及ぼすことになる。

【0028】

スプリットブレインを起きにくくするためには、故障の誤検出を起きにくくする必要がある。そのためには、ハートビートのタイムアウトを十分に長くする必要がある。実用上は、10秒～1分ぐらいのタイムアウト値が使われるのが一般的である。

【0029】

(2) 故障発生時の処理のリアルタイム性

ところが、タイムアウトを長く設定すると、故障の発生から故障検出までの時間が長くなることになる。すると、その間は、整列マルチキャストの Protokol 中で、故障したコンピュータの故障検出を待ち、整列マルチキャストの実行が一時的に停止する。その結果、多重化の実行が一時的に停止することになる。

【0030】

これは、一般的にはシステムに致命的な影響を与えるものではないが、リアル

タイム性が重要なシステムでは、故障発生時にその要件を満たさなくなる場合もある。つまり、ハートビートのタイムアウト値は、リアルタイム性の要件から上限が抑えられており、むやみに長く設定できない。

## 【0031】

結局、このハートビートのタイムアウト値の設定は、スプリットブレインとリアルタイム性の間でトレードオフの関係に陥ってしまうという問題があった。

## 【0032】

この発明は、このような事情を考慮してなされたものであり、スプリットブレインの防止と故障発生時におけるリアルタイム性の確保とを両立させることを可能とした分散システムおよび同システムの多重化制御方法を提供することを目的とする。

## 【0033】

## 【課題を解決するための手段】

前述した目的を達成するために、この発明は、故障検出をまったく行わないことによって、スプリットブレインを原理的に発生させず、タイムアウトによる故障発生時の処理の中断も発生させないようにしたものである。そして、そのために、この発明は、少なくとも $(n-f)$ 台のコンピュータが動作していれば、他の $f$ 台の動作に関わらず、入力をそれらに配送するようにした。

## 【0034】

より具体的には、この発明は、ネットワークで接続された $n$ 台のコンピュータを同期的に動作させる分散システムであって、少なくとも $(n-f)$ 台以上での多重化を保証する分散システムにおいて、前記コンピュータそれぞれは、各々が次に処理する候補として選択した入力データを前記ネットワークを介して収集する入力候補収集手段と、前記入力候補収集手段により収集された入力データが $(n-f)$ 個以上存在する場合に、その中に同一内容の入力データが $(n-f)$ 個以上あるか否かを判定し、 $(n-f)$ 個以上あったときに、その入力データを次に処理する対象として確定する第1の入力候補選定制御手段と、前記収集された入力データの中に同一内容の入力データが $(n-f)$ 個以上なかったときに、前記収集された入力データ数の過半数を占める同一内容の入力データが存在するか

否かを判定し、存在したときに、その入力データを自候補とするとともにそれ以外の他の候補の入力データをすべて破棄した上で前記入力候補収集手段に入力データの収集を再実行させる第2の入力候補選定制御手段と、前記収集された入力データ数の過半数を占める同一内容の入力データが存在しなかったときに、前記収集された入力データの中からいずれかの入力データを任意に選択して自候補とするとともに、それ以外の他の候補の入力データをすべて破棄した上で前記入力候補収集手段に入力データの収集を再実行させる第3の入力候補選定制御手段とを具備することを特徴とする分散システムを提供する。

## 【0035】

この分散システムにおいては、故障検出をまったく行わないで、整列マルチキャストを実現し、特に、故障発生時でも配送の中断を発生させることがない。

## 【0036】

## 【発明の実施の形態】

以下、図面を参照してこの発明の一実施形態を説明する。

## 【0037】

まず、この実施形態に係る分散システムの前提条件を説明する。ここでは、多重化を構成するコンピュータの数を $n$ とし、 $f$ 台までの故障停止が許容されるものと想定する。つまり、多重化されるプログラムは、少なくとも $(n-f)$ 台のコンピュータ上で実行される。また、 $(f+1)$ 台以上の故障停止が発生した場合には、多重化は継続しないものとする（いわゆるフェイルストップ）。

## 【0038】

また、ここでは、この $f$ を $3f < n$ となる最大の整数とする。たとえば $n=4$ ならば、 $f=1$ である。 $n=10$ ならば $f=3$ である。この前提は、システムの稼働率に制限を設けるものであるが、たとえば、 $n=10$ の場合には、前述の稼働率の計算より、実用上まったく問題ないといえる。

## 【0039】

また、多重化されるプログラムの入力および出力は、信頼性のないデータグラム (Unreliable Datagram) のセマンティックスであるとする。これは、入出力のパケットについて、欠落、重複および順序の交換を許容するものである。信頼

性のないデータグラムのセマンティクスをもつ例としては、I P (Internet Protocol) が挙げられる。

## 【 0 0 4 0 】

この信頼性のないデータグラムのセマンティクスにおける非決定性と多重化されるプログラムの決定性とは矛盾するものではないことに注意する。プログラムの決定性は、入力が決まれば一意的に次の状態と出力が決まることを示し、プログラムの内部動作に関する決定性を意味している。一方、信頼性のないデータグラムのセマンティクスは、あるプログラムの出力が他のプログラムの入力へ渡される途中で、欠落、重複または順序の交換があり得ることを示し、プログラム間の入出力に関する非決定性を意味している。

## 【 0 0 4 1 】

次に、図 1 および図 2 を参照して、この分散システムの構成を説明する。

## 【 0 0 4 2 】

図 1 に示すように、この分散システム 1 0 0 0 は、n 台のコンピュータ 1 0 0 により多重化されており、それぞれのコンピュータ 1 0 0 が、外部ネットワーク A を介して複数のクライアント装置 2 0 0 0 と接続されている。また、このコンピュータ 1 0 0 間は、内部ネットワーク B を介して接続されている。そして、この分散システム 1 0 0 0 における各コンピュータ 1 0 0 は、外部ネットワーク A を介してクライアント装置 2 0 0 0 から受け取った入力パケット（入力 1）または内部ネットワーク B を介して他のコンピュータ 1 0 0 から受け取った入力パケット（入力 2）を他のコンピュータ 1 0 0 と同じ順序で処理していく。

## 【 0 0 4 3 】

なお、この処理により生成される出力パケットは、外部ネットワーク A を介してクライアント装置 2 0 0 0 に返却され（出力 1）、または、内部ネットワーク B を介して他のコンピュータ 1 0 0 に転送される。

## 【 0 0 4 4 】

図 2 は、コンピュータ 1 0 0 の構成を示す図である。入力受付キュー部 1 で受け付けられた入力パケットは、整列マルチキャスト部 2 によってアプリケーションプログラム 3 に配送されることになる。配送された入力パケットの入力によっ

て、このアプリケーションプログラム3は、プログラム状態管理部4に保存されている状態にしたがって実行し、出力パケットを生成する。出力パケットは、出力フィルター部5で選別されてから出力される。

## 【0045】

次に、整列マルチキャスト部2の各構成要素について説明する。

## 【0046】

入力順序番号記憶部21は、整列マルチキャストによってそのコンピュータへ次に配送される入力パケットの順序番号を記憶する。入力パケットジャーナル記憶部22は、整列マルチキャストによってそのコンピュータへ配送が確定した入力パケットの列を最近のものから一定の量だけ記憶する。プロトコルデータ送受信部23は、他のコンピュータのプロトコルデータ送受信部23とプロトコルデータをやり取りする。

## 【0047】

また、ステップ番号記憶部24、候補パケット記憶部25および入力パケット確定判定部26は、整列マルチキャストによってそのコンピュータへ次に配送される入力パケットを決定するアルゴリズムで用いられる。ステップ番号記憶部24は、プロトコルのステップ番号を記憶する。候補パケット記憶部25は、そのステップにおける各コンピュータの「入力候補」となる入力パケットを計 $n$ 個記憶する。入力パケット確定判定部26は、候補パケット記憶部25の情報から入力パケットの確定の判定および次ステップの「入力候補」の決定を行う。

## 【0048】

最大確定入力順序番号記憶部27は、他のコンピュータも含め、配送が確定したことがわかっている最大の入力順序番号を記憶する。遅延記憶部28は、 $(n-1)$ 個のフラグで構成され、他コンピュータから遅延しているかどうかを記憶する。そして、スキップ判定部29は、遅延記憶部28の情報からスキップ動作の必要性を判定、実行する。

## 【0049】

以降、該当入力順序番号とは、入力順序番号記憶部21に記憶された入力順序番号のことを指し、該当ステップ番号とは、ステップ番号記憶部24に記憶され



たステップ番号のことを指し、該当最大確定入力順序番号とは、最大確定入力順序番号記憶部 27 に記憶された入力順序番号のことを指し、自候補とは、候補パケット記憶部 25 における自コンピュータに対応する「入力候補」を指し、他候補とは、候補パケット記憶部 25 における自候補以外の「入力候補」を指すものとする。

【0050】

図 3 は、プロトコルデータ送受信部 23 によって送受信されるプロトコルデータのレイアウトを示す図である。

【0051】

図 3 に示すように、プロトコルデータ送受信部 23 によって送受信されるプロトコルデータは、種類、送信者、入力順序番号、ステップ番号、最大確定入力順序番号および入力パケットの各フィールドを含んでいる。そして、先頭の種類フィールドによって、このプロトコルデータは、次の 3 つに使い分けられる。

【0052】

(1) 候補種類：入力順序番号フィールド、ステップ番号フィールド、入力パケットフィールドには、それぞれ、送信者の送信時における、該当入力順序番号、該当ステップ番号、自候補が格納される。

【0053】

(2) 確定種類：その入力順序番号に対応する入力パケットが、送信者の送信時における入力パケットジャーナル記憶部 26 にあることを示し、入力パケットフィールドにはその入力パケットが格納される。この場合、ステップ番号フィールドは使用しない。

【0054】

(3) 遅延種類：その入力順序番号に対応する入力パケットが、送信者の送信時における入力パケットジャーナル記憶部 26 にないことを示す。この場合、ステップ番号フィールド、入力パケットフィールドは使用しない。

【0055】

いずれの種類においても、最大確定入力順序番号フィールドには、送信者の送信時における該当最大確定入力順序番号を格納する。また、該当最大確定入力順

序番号は、そのコンピュータで確定した入力パケットの順序番号と、受信したプロトコルデータ中の最大確定入力順序番号とのうち、最も大きいものに更新するものとする。

## 【0056】

ここで、図4を参照して、整列マルチキャスト部2によって実行される整列マルチキャストの主要部の概要について説明する。

## 【0057】

いま、多重化を構成するコンピュータの数、つまり $n$ を4とする。また、前述したように、 $f$ は $3 < f < n$ となる最大の整数であるから、 $f = 1$ となる。したがって、この例では、少なくとも $(n - f)$ 、つまり3台以上で一貫性を保ちながら処理を実行していくことになる。

## 【0058】

第1に、コンピュータ(1)、(2)はA、コンピュータ(3)はB、コンピュータ(4)はCをそれぞれ入力候補として選択したとする。また、第2に、コンピュータ(1)は、コンピュータ(2)の入力候補Aとコンピュータ(3)の入力候補Bを収集したとする。つまり、コンピュータ(1)は、自候補および他候補を $(n - f)$ 個収集したことになる。この時、コンピュータ(1)は、コンピュータ(4)の入力候補の収集を待たずに、入力候補の判定を試みる。しかしながら、その中に $(n - f)$ 個の同一の候補は存在しないことから、コンピュータ(1)は、入力候補の再選択を実行する。再選択は、収集された入力候補数の過半数を占める候補があればその候補を選択し、なければその中からランダムに選択する。ここではAが過半数を占めるので、コンピュータ(1)は、第3に、Aを自候補として再選択する。

## 【0059】

この要領で、コンピュータ(2)は、コンピュータ(1)の入力候補Aとコンピュータ(4)の入力候補Cを収集した後、Aを自候補として再選択し、コンピュータ(3)は、コンピュータ(2)の入力候補Aとコンピュータ(4)の入力候補Cを収集した後、Cを自候補として再選択し、コンピュータ(4)は、コンピュータ(1)の入力候補Aとコンピュータ(2)の入力候補Aを収集した後、

Aを自候補として再選択したとする。

【0060】

第4に、コンピュータ(1)は、コンピュータ(2)の入力候補Aとコンピュータ(4)の入力候補Aを収集したとする。つまり、コンピュータ(1)は、再度、自候補および他候補を $(n-f)$ 個収集したことになる。この時、コンピュータ(1)は、コンピュータ(3)の入力候補の収集を待たずに、入力候補の判定を試みる。そして、ここでは、 $(n-f)$ 個のAが存在するため、第5に、コンピュータ(1)は、入力をAに決定する。

【0061】

一方、コンピュータ(2)は、コンピュータ(1)の入力候補Aとコンピュータ(3)の入力候補Cを収集したとする。しかしながら、 $(n-f)$ 個の同一の候補は依然として存在しないことから、コンピュータ(2)は、入力候補の再選択を実行し、その中の過半数を占めるAを自候補として選択する。同様に、コンピュータ(3)は、コンピュータ(1)の入力候補Aとコンピュータ(2)の入力候補Aを収集した後、コンピュータ(4)は、コンピュータ(2)の入力候補Aとコンピュータ(3)の入力候補Aを収集した後、それぞれAを自候補として再選択したとする。

【0062】

第6に、コンピュータ(2)は、コンピュータ(1)の入力候補Aとコンピュータ(3)の入力候補Aを収集したとする。ここでのコンピュータ(1)の入力候補Aは、既に候補ではなく確定済みの入力であるため、第7に、コンピュータ(2)は、入力をAに決定する。

【0063】

一方、コンピュータ(3)は、コンピュータ(2)の入力候補Aとコンピュータ(4)の入力候補Aを収集し、コンピュータ(4)は、コンピュータ(2)の入力候補Aとコンピュータ(3)の入力候補Aを収集したとする。そして、ここでは、双方とも $(n-f)$ 個のAが存在するため、コンピュータ(3)、(4)は、入力をAに決定する。

【0064】

つまり、この分散システムは、従来のように、各コンピュータがハートビートのやり取りによって他のコンピュータとの間で正常稼働を確認し合うようなことを一切行わないことにより、スプリットブレインを原理的に発生させず、タイムアウトによる故障発生時の処理の中断も発生させないようにし、かつ、少なくとも  $(n-f)$  台以上のコンピュータによる多重化を保証する。

## 【0065】

次に、整列マルチキャスト部2の動作原理について具体的に説明する。

## 【0066】

まず、初期状態として、入力順序番号記憶部21は初期入力順序番号（たとえば1）を記憶する。入力パケットジャーナル記憶部22は空の状態であり、ステップ番号記憶部24は初期ステップ番号（たとえば1）を記憶する。また、候補パケット記憶部25も空の状態であり、最大確定入力順序番号記憶部27は初期入力順序番号を記憶し、さらに、遅延記憶部28のすべてのフラグはリセットされている。

## 【0067】

そして、この整列マルチキャスト部2が実行する整列マルチキャストによって各コンピュータへ配送される入力パケットを決定するアルゴリズムの概要は次のようになる。

## 【0068】

## （アルゴリズム1）

該当ステップ番号が初期ステップ番号である場合に、入力受付キュー部1に入力パケットがあれば、該当ステップ番号を次に進め、自候補をその入力パケットにし、他候補を空にし、候補種類のプロトコルデータを他のすべてのコンピュータに送信する。

## 【0069】

## （アルゴリズム2）

該当入力順序番号に一致する入力順序番号を持つ候補種類のプロトコルデータを受信した場合で、そのプロトコルデータが該当ステップ番号より大きいステップ番号を持つ場合、該当ステップ番号をそのステップ番号にし、自候補および送

信者に対応する他候補をプロトコルデータ中の入力パケットにし、それら以外の他候補を空にし、候補種類のプロトコルデータを他のすべてのコンピュータに送信する。

## 【0070】

## (アルゴリズム3)

該当入力順序番号に一致する入力順序番号を持つ候補種類のプロトコルデータを受信した場合で、そのプロトコルデータが該当ステップ番号と等しいステップ番号を持つ場合、送信者に対応する他候補をプロトコルデータ中の入力パケットにする。

## 【0071】

## (アルゴリズム4)

候補パケット記憶部25における空でない「入力候補」が $(n-f)$ 個以上あるとき、入力パケット確定判定部26は次の動作をする。

## 【0072】

もし、 $(n-f)$ 個以上の同一内容の「入力候補」があれば、それを該当入力順序番号における入力パケットとして確定し、入力パケットジャーナル記憶部22に記憶し、入力受付キュー部1にそれがあれば削除し、アプリケーションプログラム3に配送し、該当入力順序番号を次に進め、該当ステップ番号を初期ステップ番号にし、候補パケット記憶部25を空にし、遅延記憶部28のすべてのフラグをリセットする。

## 【0073】

それ以外で、もし、候補パケット記憶部25の中で過半数以上の同一内容の「入力候補」があれば、該当ステップ番号を次に進め、候補パケット記憶部25における自候補をその入力パケットにし、他候補を空にし、候補種類のプロトコルデータを他のすべてのコンピュータに送信する。

## 【0074】

さらに、それ以外であれば、候補パケット記憶部25の中からランダムに入力パケットを選択し、該当ステップ番号を次に進め、候補パケット記憶部25における自候補をその入力パケットにし、他候補を空にし、候補種類のプロトコルデ

ータを他のすべてのコンピュータに送信する。

【0075】

(アルゴリズム5)

該当入力順序番号より小さい入力順序番号を持つ候補種類のプロトコルデータを受信した場合で、その入力順序番号に対応する入力データが入力パケットジャーナル記憶部22にある場合、確定種類のプロトコルデータを送信者のコンピュータに返信する。

【0076】

(アルゴリズム6)

該当入力順序番号に一致する入力順序番号を持つ確定種類のプロトコルデータを受信した場合、それを該当入力順序番号における入力パケットとして確定し、入力パケットジャーナル記憶部26に記憶し、入力受付キュー部1にそれがあれば削除し、アプリケーションプログラム3に配送し、該当入力順序番号を次に進め、該当ステップ番号を初期ステップ番号にし、候補パケット記憶部を空にし、遅延記憶部28のすべてのフラグをリセットする。

【0077】

(アルゴリズム7)

該当入力順序番号より小さい入力順序番号を持つ候補種類のプロトコルデータを受信した場合で、その入力順序番号に対応する入力データが入力パケットジャーナル記憶部22にない場合、遅延種類のプロトコルデータを送信者のコンピュータに返信する。

【0078】

(アルゴリズム8)

該当入力順序番号に一致する入力順序番号を持つ遅延種類のプロトコルデータを受信した場合に、遅延記憶部28における送信者に対応するフラグをセットする。

【0079】

(アルゴリズム9)

遅延記憶部28においてフラグがたっている数と、それ以外で候補パケット記

憶部 25 における空でない入力候補数の和が  $(n-f)$  以上であるときで、候補  
パケット記憶部 25 における空でない入力候補数が  $(n-f)$  個未満であるとき  
に、スキップ判定部 29 は、以下のスキップ動作を行う。

## 【0080】

スキップ動作は、該当入力順序番号を該当最大確定入力順序番号にし、該当ス  
テップ番号を初期ステップ番号にし、候補パケット記憶部 25 を空にし、遅延記  
憶部 28 のすべてのフラグをリセットし、プログラム状態管理部 4 にスキップを  
通知する。

## 【0081】

なお、以上の（アルゴリズム 1）～（アルゴリズム 9）の順序は、必ずしもこ  
の順序で実行されるというものではない。つまり、これらは、その条件が成立す  
れば独立して実行されるものである。

## 【0082】

また、プログラム状態管理部 4 は、スキップが通知されると、該当入力順序番  
号の直前の状態を他のコンピュータのプログラム状態管理部 4 からコピーする。  
このために、プログラム状態管理部 4 は、各入力順序番号の直前の状態を最近の  
ものから一定の量だけ保持している。

## 【0083】

ここで、上述したアルゴリズムの動作の概要を説明しながら、このアルゴリズム  
の有効性を証明する。

## 【0084】

（アルゴリズム 1）～（アルゴリズム 4）は、整列マルチキャストの 1 回の配  
送を行う基本的な部分である。従来では、故障していない全コンピュータで一致  
するまで繰り返していたのに対して、この分散システムでは、 $(n-f)$  台で一  
致するまで繰り返す。

## 【0085】

また、（アルゴリズム 5）～（アルゴリズム 6）は、短い多重化実行の遅延を  
解消するため、すでに確定している入力パケットを回送するものである。

## 【0086】

そして、(アルゴリズム7)～(アルゴリズム9)は、長い多重化実行の遅延を一足飛びに解消するため、スキップ動作を行うものである。

【0087】

まず、(アルゴリズム1)～(アルゴリズム6)が整列マルチキャストの要件を満たすことを説明する。これには、各入力順序番号で同一の入力パッケージが確定されることを示せばよい。

【0088】

入力パッケージを確定するのは、(アルゴリズム4)か(アルゴリズム6)であるが、(アルゴリズム6)の場合は、確定した入力パッケージを回送したもので、最初に(アルゴリズム4)によって入力パッケージを確定したコンピュータが必ず存在する。確定した時の入力パッケージをP、ステップ番号をSとする。

【0089】

まず、ステップS+1では、すべてのコンピュータで「入力候補」はP以外にはあり得ないことを示す。

【0090】

自分の「入力候補」を決定するのは、(アルゴリズム1)、(アルゴリズム2)または(アルゴリズム4)であるが、ステップ番号Sは初期ステップ番号ではあり得ないので、ステップS+1での「入力候補」は、(アルゴリズム2)か(アルゴリズム4)で決定される。(アルゴリズム2)は「入力候補」を回送したものであるため、結局、(アルゴリズム4)で決定するステップS+1での「入力候補」がP以外にはあり得ないことを示せばよい。

【0091】

ステップS+1での「入力候補」を(アルゴリズム4)で決定するには、ステップSでの「入力候補」が $(n-f)$ 個必要である。この集合をXとする。一方、ステップSでは、(アルゴリズム4)によって入力パッケージを確定したコンピュータがあるのだから、少なくとも $(n-f)$ 個の「入力候補」がPである。この集合をYとする。すると、

Xの要素数 $\geq n-f$

Yの要素数 $\geq n-f$



$X \cup Y$ の要素数  $\leq n$

$$X \text{の要素数} - X \cap Y \text{の要素数} = X \cup Y \text{の要素数} - Y \text{の要素数} \leq n - (n - f) = f$$

となり、 $X$ のうち $P$ でないのは、多くとも $f$ 個しかない。後は、 $f$ が $X$ の中で半数未満であることがいえれば、 $X$ の中で $P$ が過半数を占めることになり、(アルゴリズム4)によって $P$ に決定することがわかる。ここで、

$$X \text{の要素数} - 2f \geq (n - f) - 2f = n - 3f$$

となり、前述の通り、 $n - 3f > 0$ であるから、これが証明される。

【0092】

結局、ステップ $S+1$ では、すべてのコンピュータで「入力候補」は $P$ 以外にはあり得ないのだから、この入力順序番号で確定するとすれば、必ず $P$ で確定することになる。これで、整列マルチキャストの要件を満たすことが言えた。

【0093】

次に、(アルゴリズム5)～(アルゴリズム9)で行う遅延の解消について説明する。

【0094】

この遅延は、 $(n - f)$ 台よりも多い台数で多重化を実行している場合に発生する。遅延しているコンピュータは、その時点では多重化として不要であるが、進んでいるコンピュータが故障停止した場合などに、多重化を継続するために必要になる。つまり、その場合には、遅延しているコンピュータは、最終入力順序番号まで追いつかなければならない。

【0095】

(アルゴリズム5)～(アルゴリズム6)で行う短い多重化実行の遅延の解消は、単純に、進んでいるコンピュータで確定した入力パケットを回送する。入力パケットの到着順序は同じになるので、整列マルチキャストの要件は満たされている。

【0096】

一方、(アルゴリズム7)～(アルゴリズム9)で行う長い多重化実行の遅延の解消は、いわゆる「おいてけぼり」の概念を用いる。「おいてけぼり」は、進

んでいるコンピュータが確定した入力パケットを忘れてしまうほど長く遅延したときに発生する。そして、この「おいてけぼり」が判定されると、スキップ動作が行われる。スキップ動作では、入力順序番号をスキップするので、入力パケットの系列が中抜けになり、整列マルチキャストの要件を満たさなくなる。

【0097】

そこで、この中抜けになった入力パケットの系列を補うため、プログラム状態管理部4により一致化コピーを行う。これによって、多重化は矛盾なく続行することができる。

【0098】

次に、信頼性のないデータグラムのセマンティクスとの関係に触れる。

【0099】

出力に関しては、信頼性のないデータグラムのセマンティクスなので、出力フィルタ部5の動作は任意でよい。たとえば、無選別で出力すると、出力パケットが多重化を実行するコンピュータの数だけ出力されることになるが、信頼性のないデータグラムのセマンティクスでは、パケットの重複を許すので、この範囲内である。

【0100】

また、この分散システムでは、多重化実行の遅延が発生するため、特に出力パケットに関して順序の交換が発生する可能性がある。これは、進んでいるコンピュータが出力した後、遅延しているコンピュータが意味的にはそれ以前の出力を実行するためである。

【0101】

しかしながら、性能面などにおいて、出力フィルタ部5の設定は重要であり、たとえば、(アルゴリズム4)で入力パケットが確定したときは、出力フィルタを開、(アルゴリズム6)で入力パケットが確定したときは、出力フィルタを閉と設定すれば、順序の交換を低減することができる。また、(アルゴリズム4)で入力パケットが確定し、その入力パケットが入力受付キュー1から取り除かれた場合にのみ、出力フィルタを開、それ以外では閉とすれば、重複を低減することができる。

## 【0102】

すなわち、この分散システムは、すくなくとも  $n-f$  台のコンピュータが動作していれば、他の  $f$  台の動作に関係なく入力をそれらに配送することにより、整列マルチキャストを故障検出を使わないで実現し、特に、故障発生時でも、配送の中断が発生しない。

## 【0103】

また、最大で  $f$  台のコンピュータで、プログラムの多重化の実行が遅延する可能性があることを考慮し、この遅延された実行がスプリットブレインを起こさないように追い付く仕組みを実現する。

## 【0104】

次に、図5乃至図10を参照して、整列マルチキャスト部2の動作手順について説明する。

## 【0105】

図5および図6は、整列マルチキャストの1回の配送を行う基本的な部分の動作手順を示すフローチャートである。

## 【0106】

整列マルチキャスト部2は、まず、候補一覧作成処理を実行する（図5のステップA1）。この候補一覧作成は、該当ステップ番号が初期値のときは（図6のステップB1のYES）、受付キューに入力パケットが存在するかどうかを調べて（図6のステップB2）、存在すれば（図6のステップB2のYES）、該当ステップ番号を次に進め（図6のステップB3）、受付キューの入力パケットを自候補とし、かつ、この自候補を他のすべてのコンピュータに送信する（図6のステップB4）。

## 【0107】

一方、該当ステップ番号が初期値でないか（図6のステップB1のNO）、または受付キューに入力パケットがないとき（図6のステップB2のNO）、整列マルチキャスト部2は、同一の入力順序番号を持つプロトコルデータを受信しているかどうか判定し（図6のステップB5）、受信していれば（図6のステップB5のYES）、今度は、受信データ内のステップ番号は該当ステップ番号より

も大きいかどうかを判定する（図6のステップB6）。そして、該当ステップ番号よりも大きければ（図6のステップB6のYES）、整列マルチキャスト部2は、該当ステップを受信データ内のステップ番号に更新した後（図6のステップB7）、受信データ内の入力パケットを自候補とし、かつ、この自候補を他のすべてのコンピュータに送信する（図6のステップB8）。このとき、整列マルチキャスト部2は、この入力パケットを他候補として記憶しておく。また、受信データ内のステップ番号と該当ステップ番号とが等しければ（図6のステップB6のNO、ステップB9のYES）、受信データ内の入力パケットを他候補として記憶する（図6のステップB10）。

## 【0108】

ここで、整列マルチキャスト部2は、記憶した候補数が $(n-f)$ 個以上になったかどうかを調べ（図6のステップB11）、なっていないならば（図6のステップB11のNO）、ステップB1からの処理を繰り返し、なっていれば（図6のステップB11のYES）、この処理を終了する。

## 【0109】

候補一覧作成処理が終了すると、整列マルチキャスト部2は、 $(n-f)$ 個以上の同一の候補が存在するかどうかを調べ（図5のステップA2）、存在すれば（図5のステップA2のYES）、その候補を入力パケットとして確定する（図5のステップA3）。つまり、この入力パケットを受付キューから削除するとともに、アプリケーションプログラム3に投入する。そして、整列マルチキャスト部2は、次工程へ移行すべく、入力順序番号を次に進め、該当ステップ番号を初期化し、記憶したすべての候補を破棄し、遅延フラグをリセットする（図5のステップA4）。

## 【0110】

一方、 $(n-f)$ 個以上の同一の候補が存在しなかった場合（図5のステップA2）、整列マルチキャスト部2は、今度は、過半数以上の同一の候補が存在するかどうかを調べ（図5のステップA5のYES）、存在すれば（図5のステップA5のYES）、その候補を自候補とし、かつ、この自候補を他のすべてのコンピュータに送信した上で（図5のステップA6）、ステップA1からの処理を

繰り返す。この時、整列マルチキャスト部2は、記憶していた他候補をすべて破棄する。また、過半数以上の同一の候補が存在しなければ（図5のステップA5のNO）、整列マルチキャスト部2は、ランダムに自候補を選択し、かつ、この自候補を他のすべてのコンピュータに送信した上で（図5のステップA7）、ステップA1からの処理を繰り返す。この時も、整列マルチキャスト部2は、記憶していた他候補をすべて破棄する。

## 【0111】

以上の手順で、各コンピュータは、故障検知を行わず、 $(n-f)$ 台以上の一致を確認しながら処理を進めていく。

## 【0112】

また、図7乃至図10は、多重化実行の遅延を解消するための動作手順を示すフローチャートである。

## 【0113】

整列マルチキャスト部2は、該当入力順序番号より小さい入力順序番号を持つ候補種類のプロトコルデータを受信した場合、その入力順序番号に対応する入力パケットがジャーナルに存在するかどうかを調べる（図7のステップC1）。そして、整列マルチキャスト部2は、ジャーナルに存在すれば（図7のステップC1のYES）、その入力パケットをセットした確定種類のプロトコルデータを送信者に返送し（図7のステップC2）、一方、存在しなければ（図7のステップC1のNO）、遅延種類のプロトコルデータを送信者に返送する（図7のステップC3）。

## 【0114】

また、整列マルチキャスト部2は、該当入力順序番号に一致する入力番号を持つ確定種類のプロトコルデータを受信した場合、その受信データ内の入力パケットを入力パケットとして確定する（図8のステップD1）。つまり、この入力パケットを受付キューから削除するとともに、アプリケーションプログラム3に投入する。そして、整列マルチキャスト部2は、次工程へ移行すべく、入力順序番号を次に進め、該当ステップ番号を初期化し、記憶したすべての候補を破棄し、遅延フラグをリセットする（図8のステップD2）。

## 【0115】

また、整列マルチキャスト部2は、該当入力順序番号に一致する入力順序番号を持つ遅延種類のプロトコルデータを受信した場合、送信者に対応する遅延フラグをセットする。

## 【0116】

また、整列マルチキャスト部2は、セットされた遅延フラグ数と記憶された候補数との和が $(n-f)$ 個以上に達したかどうかを監視し(図10のステップF1)、 $(n-f)$ 個以上に達していれば(図10のステップF1のYES)、その記憶された候補数が $(n-f)$ 個未満かどうかを調べる(図10のステップF2)。そして、 $(n-f)$ 個未満であれば(図10のステップF2のYES)、整列マルチキャスト部2は、スキップ動作を行なう(図10のステップF3)。つまり、該当入力順序番号を該当最大確定入力順序番号にし、該当ステップ番号を初期ステップ番号にし、候補パケット記憶部25を空にし、遅延記憶部28のすべてのフラグをリセットした上で、プログラム状態管理部4にスキップを通知する。

## 【0117】

以上の手順で、各コンピュータは、スプリットブレインを起こさないよう、遅延された実行が追い付く仕組みを実現する。

## 【0118】

なお、本発明は、上記実施形態に限定されるものではなく、実施段階ではその要旨を逸脱しない範囲で種々に変形することが可能である。更に、上記実施形態には種々の段階の発明が含まれており、開示される複数の構成要件における適宜な組み合わせにより種々の発明が抽出され得る。例えば、実施形態に示される全構成要件から幾つかの構成要件が削除されても、発明が解決しようとする課題の欄で述べた課題が解決でき、発明の効果の欄で述べられている効果が得られる場合には、この構成要件が削除された構成が発明として抽出され得る。

## 【0119】

## 【発明の効果】

以上、詳述したように、この発明によれば、 $n$ 台のコンピュータで多重化を構

成し、 $f$  台までの故障停止が許容される場合に、少なくとも  $(n - f)$  台のコンピュータが動作していれば、他の  $f$  台の動作に関わらず、入力がそれらに配送されるようになる。つまり、故障検出をまったく行わないことによって、スプリットブレインを原理的に発生させず、タイムアウトによる故障発生時の処理の中断も発生させることがない。

【0 1 2 0】

また、最大で  $f$  台のコンピュータで、プログラムの多重化の実行が遅延する可能性があることを考慮し、この遅延された実行がスプリットブレインを起こさないように追い付く仕組みも実現する。

【図面の簡単な説明】

【図 1】

この発明の実施形態に係る分散システムの構成を示す図。

【図 2】

同実施形態の分散システムを構成するコンピュータの機能ブロック図。

【図 3】

同実施形態の分散システムを構成するコンピュータ間で送受信されるプロトコルデータのレイアウトを示す図。

【図 4】

同実施形態の分散システムが実行する整列マルチキャストの主要部の概要について説明するための図。

【図 5】

同実施形態の分散システムが実行する整列マルチキャストの 1 回の配送を行う基本的な部分の動作手順を示す第 1 のフローチャート。

【図 6】

同実施形態の分散システムが実行する整列マルチキャストの 1 回の配送を行う基本的な部分の動作手順を示す第 2 のフローチャート。

【図 7】

同実施形態の分散システムが実行する、多重化実行の遅延を解消するための動作手順を示す第 1 のフローチャート。

【図 8】

同実施形態の分散システムが実行する、多重化実行の遅延を解消するための動作手順を示す第 2 のフローチャート。

【図 9】

同実施形態の分散システムが実行する、多重化実行の遅延を解消するための動作手順を示す第 3 のフローチャート。

【図 1 0】

同実施形態の分散システムが実行する、多重化実行の遅延を解消するための動作手順を示す第 4 のフローチャート。

【符号の説明】

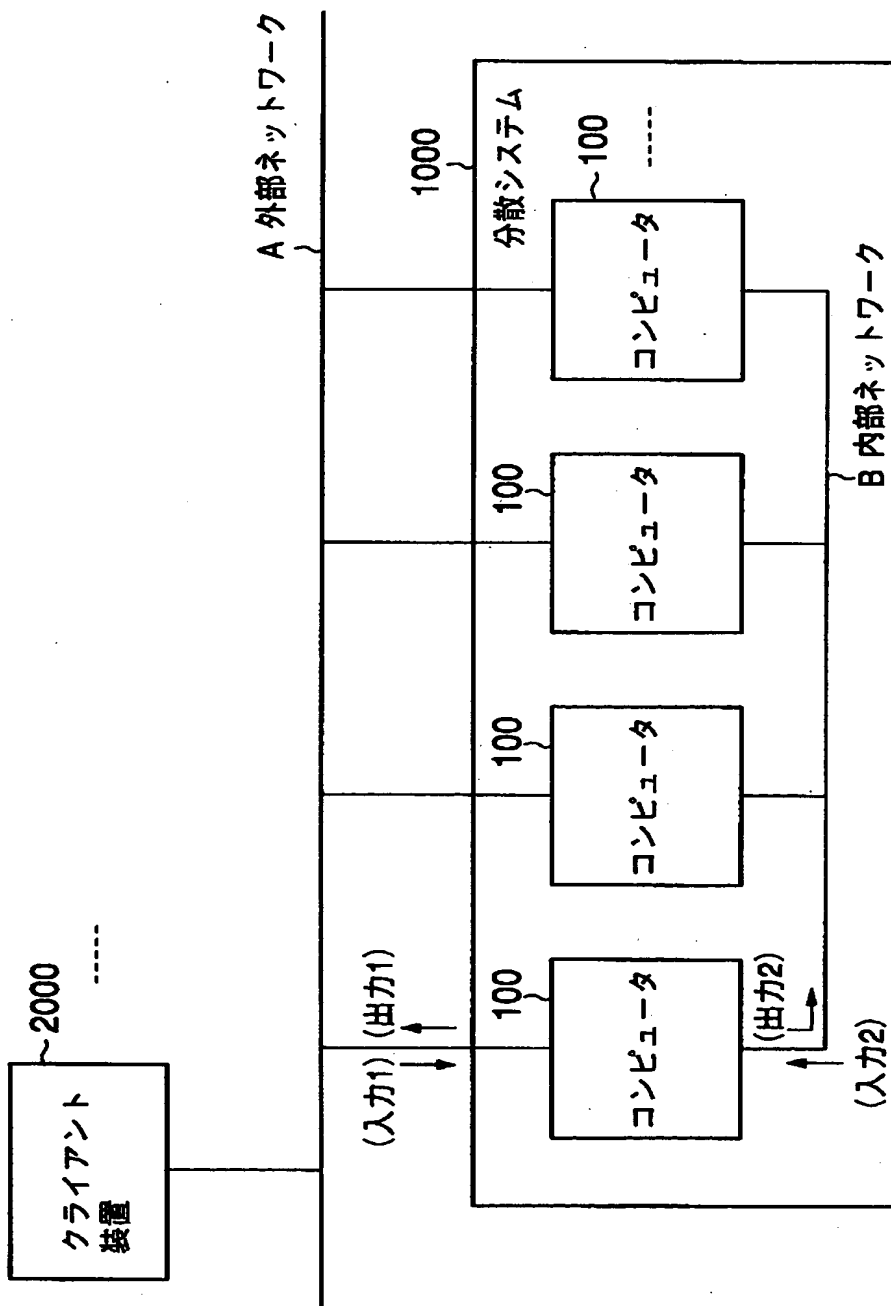
- 1 …入力受付キュー
- 2 …整列マルチキャスト部
- 3 …アプリケーションプログラム
- 4 …プログラム状態管理部
- 5 …出力フィルタ部
- 2 1 …入力順序番号記憶部
- 2 2 …入力パケットジャーナル記憶部
- 2 3 …プロトコルデータ送受信部
- 2 4 …ステップ番号記憶部
- 2 5 …候補パケット記憶部
- 2 6 …入力パケット確定判定部
- 2 7 …最大確定入力順序番号記憶部
- 2 8 …遅延記憶部
- 2 9 …スキップ判定部
- 1 0 0 …コンピュータ
- 1 0 0 0 …分散システム
- 2 0 0 0 …クライアント装置
- A …外部ネットワーク
- B …内部ネットワーク



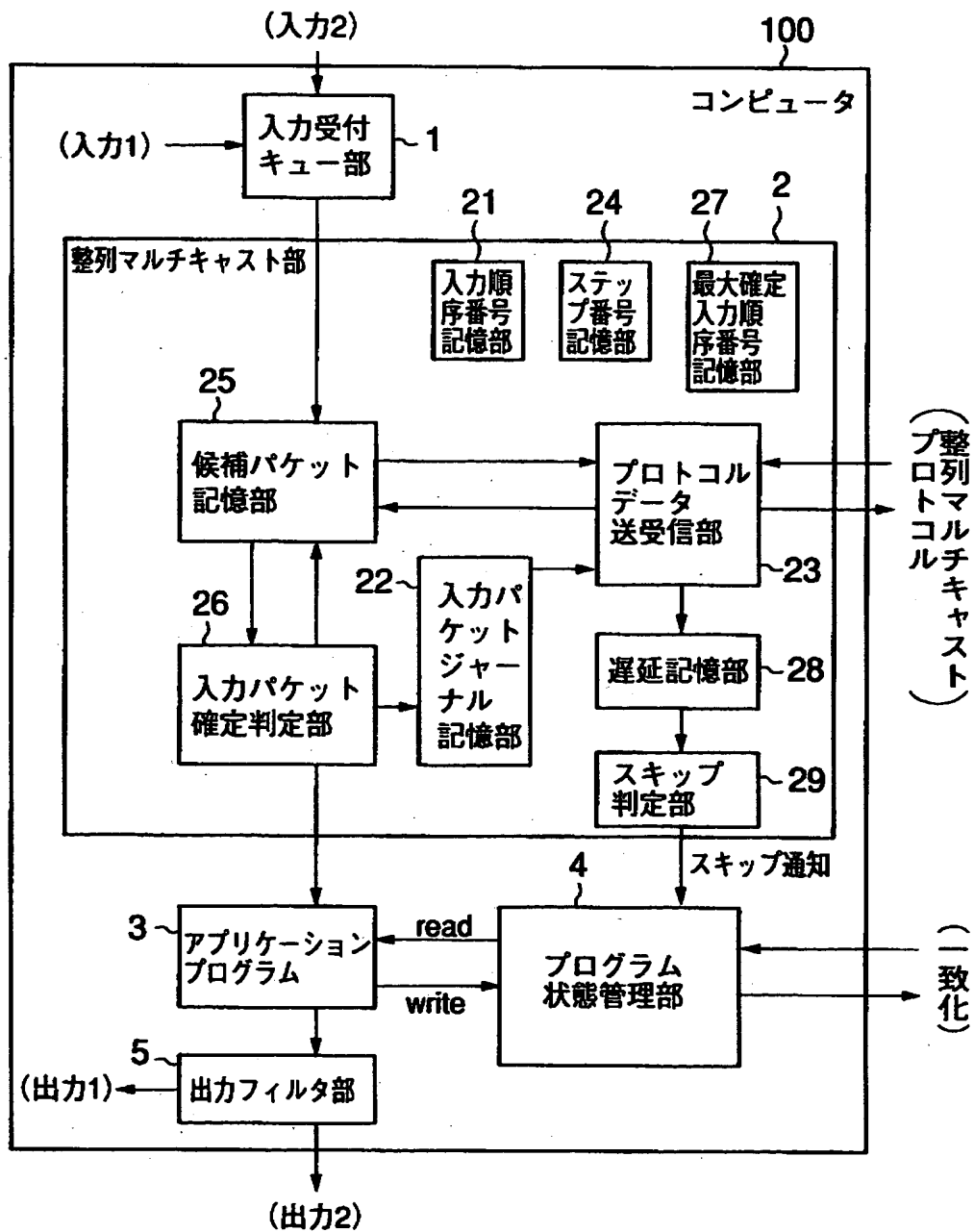
【書類名】

図面

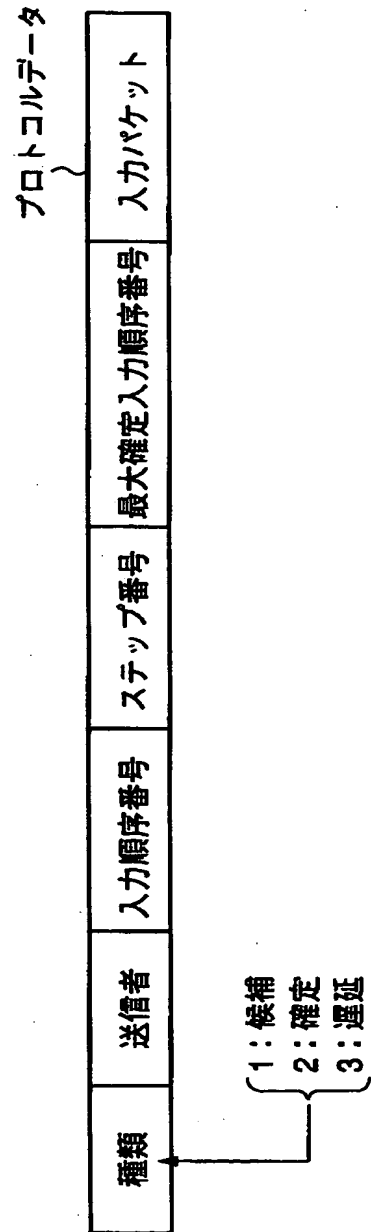
【図 1】



【図 2】



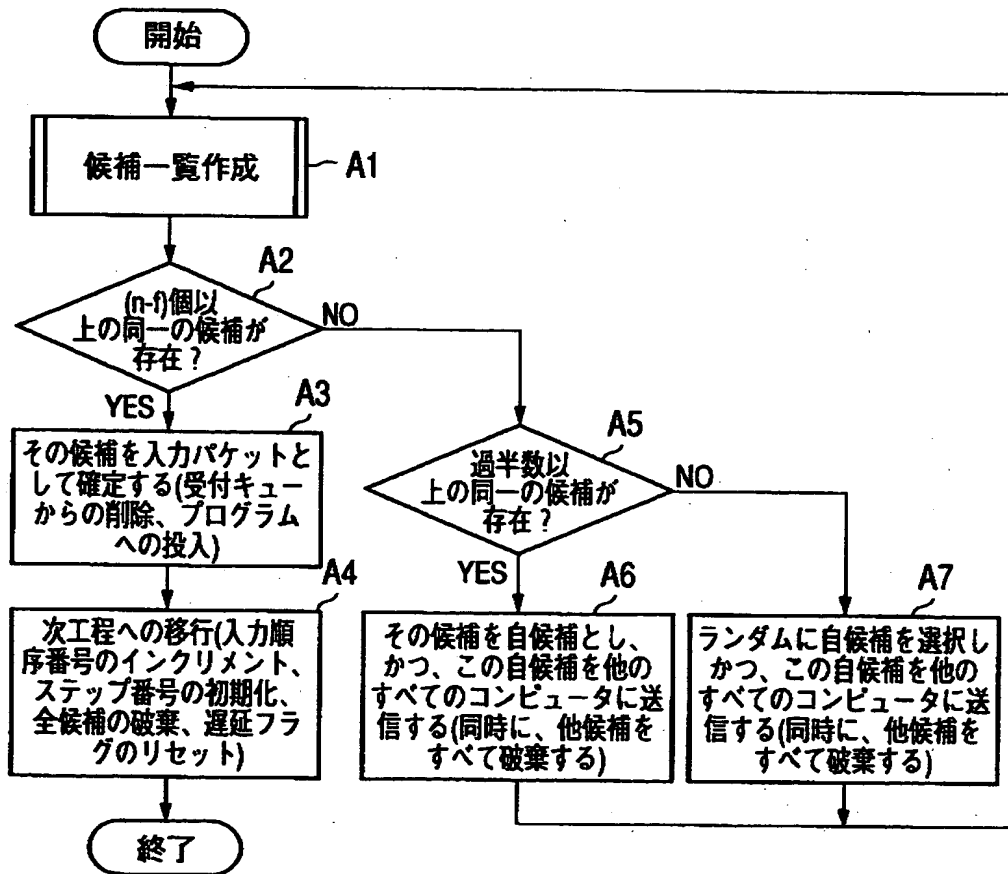
【図3】



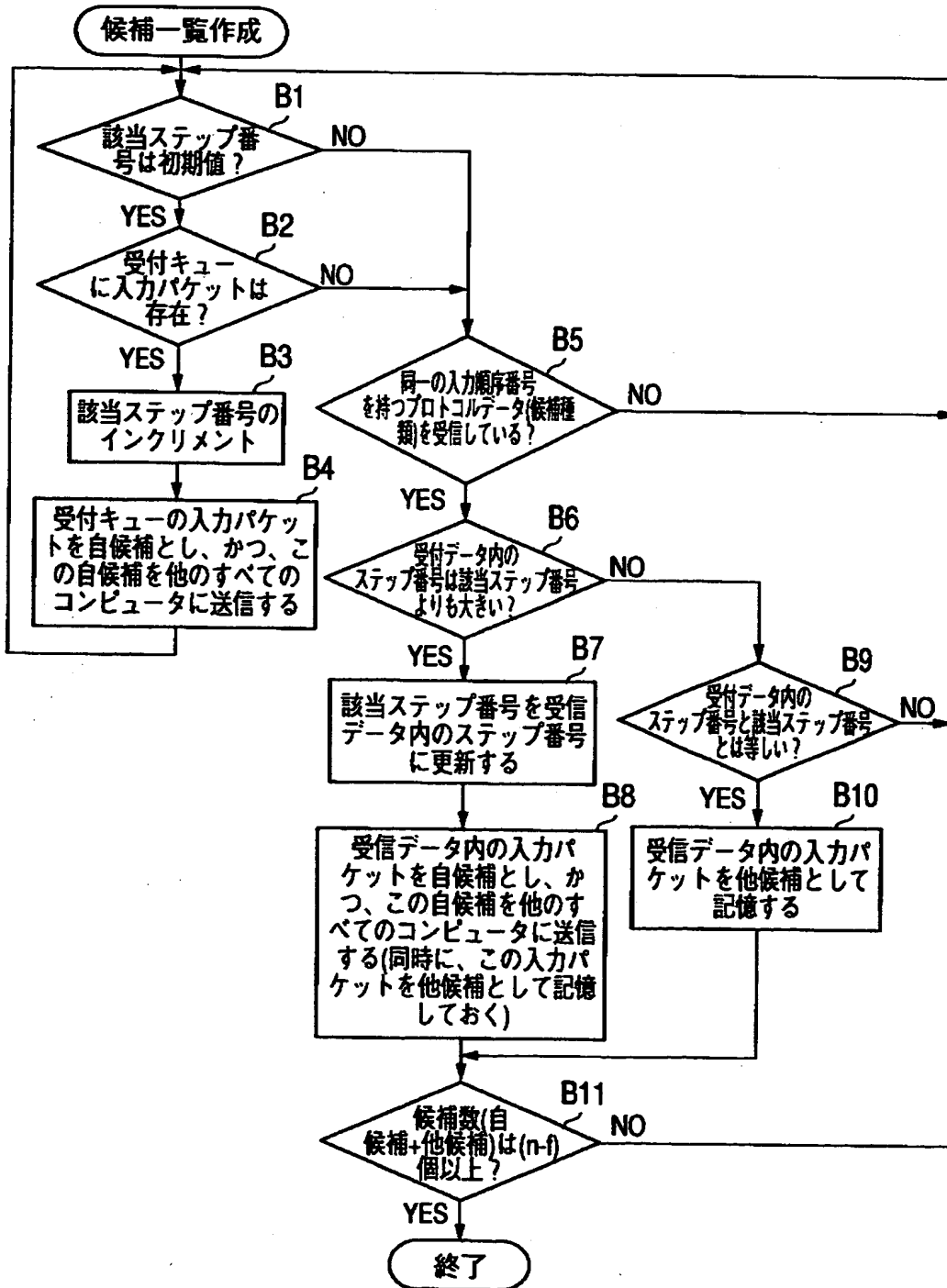
【図 4】

|   | コンピュータ(1)                    | コンピュータ(2)                    | コンピュータ(3)                    | コンピュータ(4)                    |
|---|------------------------------|------------------------------|------------------------------|------------------------------|
| 1 | A<br>(候補選択)                  | A<br>(候補選択)                  | B<br>(候補選択)                  | C<br>(候補選択)                  |
| 2 | [A(1),A(2),B(3)]<br>(候補一覧作成) | [A(1),A(2),C(4)]<br>(候補一覧作成) | [A(2),B(3),C(4)]<br>(候補一覧作成) | [A(1),A(2),C(4)]<br>(候補一覧作成) |
| 3 | A<br>(候補選択)                  | A<br>(候補選択)                  | C<br>(候補選択)                  | A<br>(候補選択)                  |
| 4 | [A(1),A(2),A(4)]<br>(候補一覧作成) | [A(1),A(2),C(3)]<br>(候補一覧作成) | [A(1),A(2),C(3)]<br>(候補一覧作成) | [A(2),C(3),A(4)]<br>(候補一覧作成) |
| 5 | 決定(A)                        | A<br>(候補選択)                  | A<br>(候補選択)                  | A<br>(候補選択)                  |
| 6 | ——                           | [A(1),A(2),A(3)]<br>(候補一覧作成) | [A(2),A(3),A(4)]<br>(候補一覧作成) | [A(2),A(3),A(4)]<br>(候補一覧作成) |
| 7 | ——                           | 決定(A)                        | 決定(A)                        | 決定(A)                        |
| ⋮ | ⋮                            | ⋮                            | ⋮                            | ⋮                            |

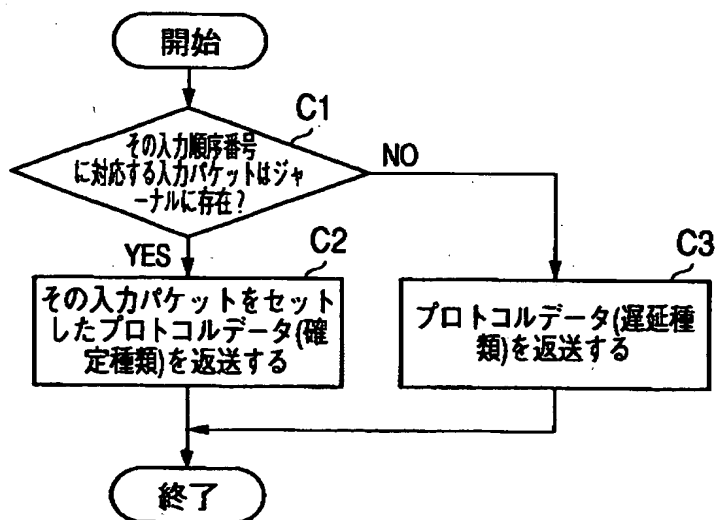
【図 5】



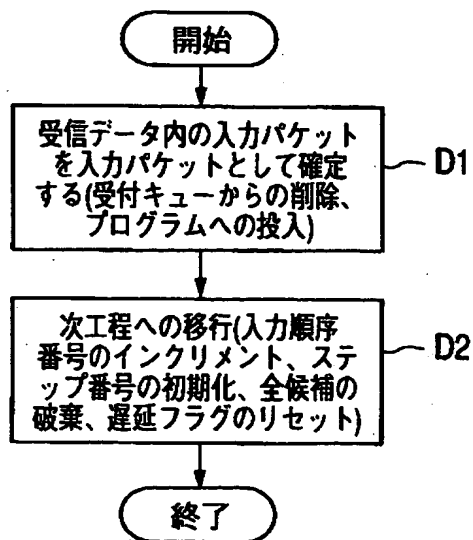
【図 6】



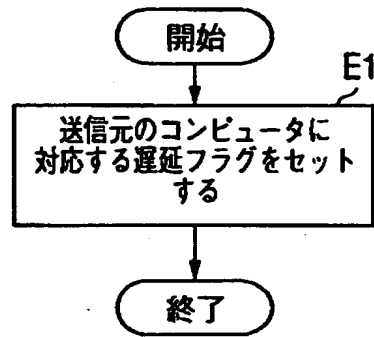
【図 7】



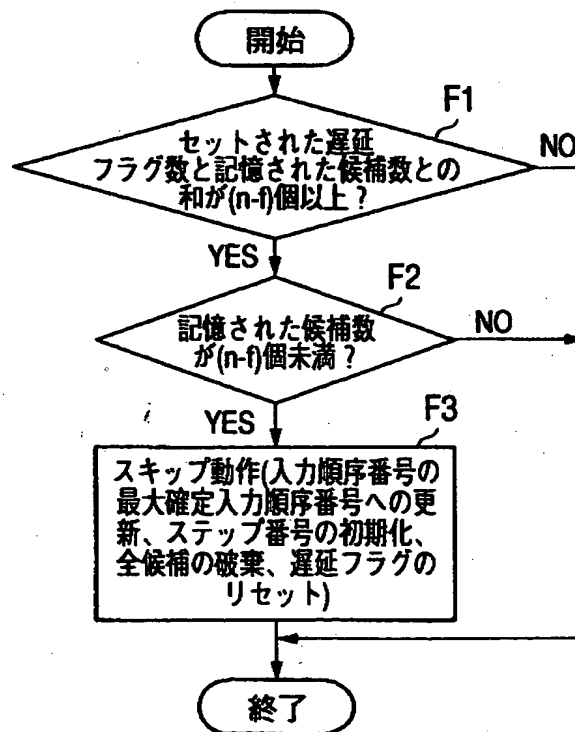
【図 8】



【図 9】



【図 10】





【書類名】            要約書

【要約】

【課題】 スプリットブレインの防止と故障発生時におけるリアルタイム性の確保とを両立させることを可能とした分散システムを提供する。

【解決手段】 この分散システムは、 $n$  台のコンピュータで多重化を構成し、 $f$  台までの故障停止を許容する。そして、各コンピュータは、内部ネットワーク  $B$  を介して入力候補を送受信し合い、その一覧を作成する。そして、その中に  $(n - f)$  個の同一の入力候補が現れるまで、それぞれがこの一覧作成を繰り返し、この条件を満たしたもののから、他のコンピュータの状態に関わらずに、その処理を実行する。つまり、この分散システムは、故障検出をまったく行わないことによって、スプリットブレインを原理的に発生させず、タイムアウトによる故障発生時の処理の中断も発生させることがない。

【選択図】        図 1

出 願 人 履 歴 情 報

識別番号 [000003078]

1. 変更年月日 1990年 8月22日  
[変更理由] 新規登録  
住 所 神奈川県川崎市幸区堀川町72番地  
氏 名 株式会社東芝
2. 変更年月日 2001年 7月 2日  
[変更理由] 住所変更  
住 所 東京都港区芝浦一丁目1番1号  
氏 名 株式会社東芝